To appear in the *AIAA Journal of Aircraft*, 35(2), 1998

Multilevel Simulation and Numerical Optimization of Complex Engineering Designs*

Mark Schwabacher†
National Institute of Standards and Technology, Gaithersburg, MD 20899

Andrew Gelsey‡
Rutgers, The State University of New Jersey, Piscataway, NJ 08855

Abstract

Multilevel representations have been studied extensively by artificial intelligence researchers. We present a general method that utilizes the multilevel paradigm to attack the problem of performing multidiscipline engineering design optimization in the presence of many local optima. The method uses a multidisciplinary simulator at multiple levels of abstraction, paired with a multilevel search space. We tested the method in the domain of conceptual design of supersonic transport aircraft, focusing on the airframe and the exhaust nozzle, and using sequential quadratic programming as the optimizer at each level. We found that using multilevel simulation and optimization can decrease the cost of design space search by an order of magnitude.

1 Introduction

A major barrier to the use of gradient-based search methods for engineering design is that complex, multidisciplinary design spaces tend to have many apparent local optima — both real and pathological. In Section 4 we define a pathological local optimum as one that the optimizer declares to be a local optimum, but that is not an optimum in the true physics of the problem, and we explore the causes of pathological local optima.

One approach to the problem of multiple local optima is to use global search methods such as genetic algorithms and simulated annealing. We would, however, like to exploit the power of gradient-based optimization methods to quickly converge on the optimum. Our general approach (described in Section 5) is therefore to use gradient-based optimization at multiple search space levels (where each level has a much smaller number of apparent local optima), coupled with multiple levels of abstraction in the simulator.

Multilevel representations have been studied extensively by artificial intelligence researchers. Multilevel techniques for planning and theorem proving go back as far as ABSTRIPS.[1] The importance of decomposing a problem into multiple levels was discussed at length in Simon's classic work on AI and Design.[2] Some researchers have applied the multilevel paradigm to engineering design,[3–6] but have not focused on the use of multilevel optimization to deal with multiple optima in the search space.

It might be asked how complex design problems are approached today. Human design-

*An earlier version of this article was presented at 6^{th} AIAA/NASA/USAF Multidisciplinary Analysis & Optimization Symposium, Bellevue, WA, September 1996. AIAA Paper No. 96-4021.

†Postdoctoral Research Associate, Engineering Design Technologies Group, Manufacturing Systems Integration Division. Member AIAA.

‡Assistant Professor, Computer Science Department. Member AIAA.

ers often decompose multidisciplinary design problems into smaller components that are then designed by different groups of people, but this entire process is generally carried out without the use of automated design. Our multilevel optimization method can therefore be seen as an automation of the approach typically taken by groups of human designers.

We tested our technique in the domain of conceptual design of supersonic aircraft, focusing on the airframe and the jet engine exhaust nozzle, and found that using multiple levels of simulation and optimization improves optimization performance by an order of magnitude.

2 Related work

Other work that uses the multilevel paradigm is described in the introduction section, but none of this work focuses on the problem of performing optimization in the presence of many local optima. Much work has been done on the use of simulated annealing and genetic algorithms to deal with search spaces with many local optima,[7–9] but none of this work has proposed the use of multiple levels of abstraction to reduce the number of apparent local optima that must be handled by the optimizer at any given time. Simulated annealing and genetic algorithms tend to be much slower than gradient-based optimization. They tend to require thousands, or even tens of thousands, of simulations and are thus not practical when each simulation is expensive.

A great deal of work has been done in the area of numerical optimization algorithms. Gill[10] provides an applications-oriented overview of numerical optimization algorithms. Peressini[11] describes the mathematics of nonlinear programming algorithms. Vanderplaats[12] describes the application of numerical optimization to engineering design. Moré and Wright[13] provide a guide to commercially available numerical optimization software. None of this literature addresses the particular difficulties of attempting to optimize functions defined by large "real-world" numerical simulators.

A number of research efforts have combined AI techniques with numerical optimization. Ellman *et al.*[14] describe a method for switching between a less expensive, less accurate simulator, and a more expensive, more accurate simulator during optimization, based on the magnitude of the gradient. Bouchard *et al.*[15] describe ways in which expert systems could be applied to the parametric design of aeronautical systems. Hoeltzel and Chieng[16] describe a system for digital chip design in which design is done at an abstract level, using machine learning to estimate the performance that would be obtained if the design were carried out at a more detailed level. Orelup *et al.*[17] describe a system called Dominic II that uses an expert system to switch among various strategies during numerical optimization. None of these efforts is focused directly on the problems of multiple local optima addressed in this article.

Powell[8,18,19] has built a module called Inter-GEN, part of the ENGINEOUS system,[20] that seeks to combine the ability of genetic algorithms to handle multiple local optima with the speed of numerical optimization algorithms. It contains a genetic algorithm, and a numerical optimizer, and uses a rule-based expert system to decide when to switch between the two. Powell has tested his system on a realistic jet engine design problem.

Gage[21,22] has also combined genetic algorithms with gradient-based optimization. He combined GA's with Sequential Quadratic Programming (SQP; see Section 3) in two ways. The first method, which he tested in the domain of aircraft wing design, first uses a GA to search a space of wing configurations that is described using a grammar, and then uses SQP to optimize the size of the wings. The second method, which he tested in the domain of truss design, uses the GA to search a space of truss configurations that is described using a grammar, while using SQP at each iteration of the GA to optimize the size of the members. Using the GA to search a configuration space before using SQP to optimize the sizes in a particular configuration can be seen as a method of search space selection which addresses the problem of multiple local optima. Further, the GA has the potential to find a smooth subspace of the overall search space before starting SQP.

Work on the use of numerical optimization in aircraft design includes that of Sobieszczanski-Sobieski *et al.*[4] and Kroo *et al.*[23] Bramlette *et al.*[24] survey the application of genetic algorithms to the design and manufacture of aeronautical systems. Sobieszczanski-Sobieski and Haftka[25] provide a survey of multidisciplinary aerospace design optimization.

3 Search Procedure

In this article we will focus on search of a space of candidate designs using numerical optimization methods which vary a set of continuous parameters to minimize a nonlinear objective function subject to a set of nonlinear equality and inequality constraints. The numerical optimizer used in the experiments described in this article is CFSQP[26] (C code for Feasible Sequential Quadratic Programming), a state-of-the-art implementation of the Sequential Quadratic Programming method. (Earlier we tried doing optimization in this domain using several different optimization packages, and found that we obtained the best results when using CFSQP.) Sequential Quadratic Programming is a quasi-Newton method that solves a nonlinear constrained optimization problem by solving a sequence of quadratic programming problems (a quadratic programming problem consists of a quadratic objective function to be optimized, and a set of linear constraints) as follows:

1. fit a quadratic programming problem to the constrained nonlinear programming problem
2. solve the quadratic programming problem
3. perform a minimization along the line defined by the current point and the minimum of the quadratic programming problem
4. repeat

4 Pathological optima

We define an *apparent local optimum* to be a point that our optimizer, CFSQP (see Section 3), declares to be a local optimum. Such a point may be a true local optimum in the true physics of the problem, or it may be a *pathological local optimum*. Pathological local optima occur for several reasons.

CFSQP terminates when one of two conditions is met. The first is that the Kuhn-Tucker conditions[27] are satisfied (within certain tolerances). The Kuhn-Tucker conditions are necessary but not sufficient for a point to be a local optimum. Further, they are based on certain assumptions about the smoothness of the objective and constraint functions, which may not hold for objective and constraint functions that are defined by realistic simulators. When these smoothness assumptions are violated, the Kuhn-Tucker conditions are neither necessary nor sufficient for a point to be a local optimum. The second condition which causes CFSQP to terminate is the failure of the line search in the direction of the minimum of the quadratic programming problem to find a point that improves the objective function while satisfying all of the constraints. This condition can occur at points in the search space where the objective or constraint functions are nonsmooth. One type of nonsmoothness — also known as "ridges" — is caused by discontinuities in the objective or constraint functions or their derivatives. "Near ridges" — portions of the search space that are smooth, but that have very large second derivatives — can similarly fool CFSQP. In the aircraft design example that we present in Section 8, the global optimum actually violates the Kuhn-Tucker conditions. CFSQP stops there because of its second termination condition, suggesting that the search space is nonsmooth at the global optimum. Interestingly, the multistart optimization (see Section 5) found about 25 other local optima, all of which satisfy the Kuhn-Tucker conditions (within a certain tolerance). The question of whether a point in a multidimensional space is a local optimum, in the absence of smoothness assumptions, is undecidable,[28] so it would not be possible for an optimizer to have a "perfect" termination criterion that only stopped at true local optima in an arbitrary space.

Another source of pathological local optima is numerical truncation error in the solvers within the simulator. These errors can result in local optima in the search space defined by the simulator that are not in fact local optima in the true physics. Apparent local optima are a barrier to the use of optimization, whether they are real or pathological.

5 The General Method

Because the search space has many apparent local optima, we use a technique that we call "random multistart" to attempt to find the global optimum. In an n-point random multistart, the engineer first chooses a box which he or she believes will contain all reasonable designs. The system randomly generates starting points within this box until it finds n evaluable points, and then performs a gradient-based optimization from each of these points. (Some randomly generated designs, which we call "unevaluable points," cannot be simulated, either because the designs are meaningless or because of limitations of the simulator.) The best design found in these n optimizations is taken to be the global optimum.

Our approach is to use random multistart gradient-based optimization at multiple search space levels, coupled with multiple levels of abstraction in the simulator. We propose two ways of creating these levels. The first is *decomposition*. The search space levels are formed by decomposing the set of design parameters into two or more subsets. These subsets will typically correspond to different components of the artifact, such as the airframe and the nozzle of an aircraft. These subsets may be disjoint, or it may be necessary for a small number of design parameters to occur in more than one subset. The most important parameters of one component may be included in another component's abstract space in order to serve as a proxy for the first component during optimization. The search space levels should be defined in such a way that the different levels are as independent as possible — that is, the optimal values in one subset should not depend strongly on the values in the other subset. If the overall space is approximately a product space of the abstract spaces, and the abstract spaces each have a moderate number of local optima, then the number of local optima in the overall space will be approximately equal to the product of the number of optima in each of the abstract spaces. Thus by decomposing the problem into multiple levels, it should be possible to optimize in search spaces with a much smaller number of local optima. For example, it may be the case that the two decomposed spaces have n and m apparent local optima, respectively, and the overall space has mn apparent local optima. If the number of multistarts needed to find the global optimum with a certain probability varies linearly with the number of apparent local optima, then the cost of having a certain probability of finding the global optimum will be $O(mn)$ in the overall space, and only $O(m+n)$ in the decomposed space.

The second method of creating the levels is *abstraction*. In this case, the levels form a hierarchy in which the earlier levels are simplified abstractions of the later levels. The earlier levels represent the same design at a lower level of detail than do the later levels.

In both cases, it is also helpful to have a simulator which can simulate at different levels of abstraction corresponding to the different levels of the search space. When optimizing one component of the overall design, it helps to have a simulator that simulates the other components at a lower level of detail. When optimizing at an abstract level, it also helps to use a simplified simulator. Such a simplified simulator can be faster and less pathological than the full simulator.

6 Aircraft Design

We have pursued our investigation in the domain of conceptual design of supersonic transport aircraft.[29] However, in our design task the key design variables have already been identified (in collaboration with an aircraft industry design expert) so a more precise characterization of our problem might be "parametric design at a system level of abstraction".

Figure 1 shows a diagram of the airframe of a typical airplane automatically designed by our software system to fly the mission shown in

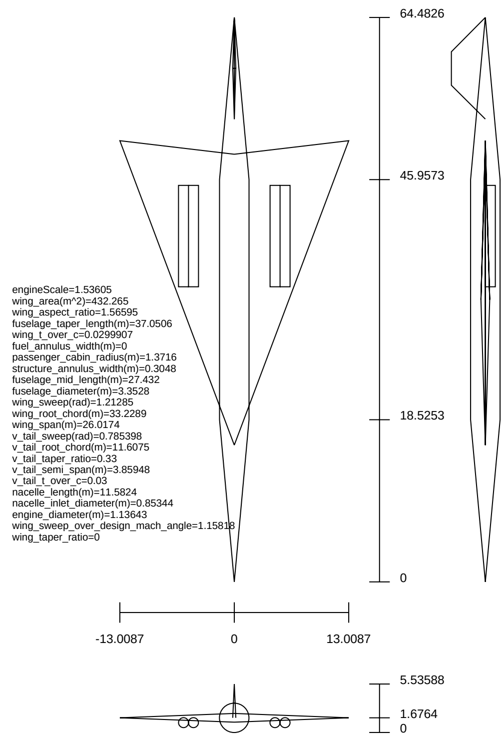

Figure 1: Supersonic transport aircraft designed by our system (dimensions in meters)

		Altitude		Duration	
Phase	Mach	m	ft	(min)	comment
1	0.227	0	0	5	"takeoff"
2	0.85	12 192	40 000	85	subsonic cruise (over land)
3	2.0	18 288	60 000	180	supersonic cruise (over ocean)

capacity: 70 passengers.

Table 1: Mission specification for aircraft in Figure 1

Table 1. This mission is for a supersonic passenger transport, so a key requirement is the passenger capacity (70 persons in this case). The mission has three key phases: a short, low-speed, ground level phase to test takeoff capability, a subsonic cruise phase representing travel over land where supersonic flight is prohibited, and finally a supersonic cruise phase corresponding to an ocean crossing.

In our system, the optimizer attempts to find a good aircraft conceptual design for a particular mission by varying major aircraft parameters such as wing area, aspect ratio, engine size, etc., using a numerical optimization algorithm. The optimizer evaluates candidate designs using a multidisciplinary simulator. In our current implementation, the design goal is to minimize the takeoff mass of the aircraft, a measure of merit commonly used in the aircraft industry at the conceptual design stage. Takeoff mass is the sum of fuel mass, which provides a rough approximation of the operating cost of the aircraft, and "dry" mass, which provides a rough approximation of the cost of building the aircraft. The simulator computes the takeoff mass of a particular aircraft design for a particular mission as follows:

1. Compute "dry" mass using historical data to estimate the weight of the aircraft as a function of the design parameters and passenger capacity required for the mission.
2. Compute the landing mass $m(t_{\text{final}})$ which is the sum of the fuel reserve plus the "dry" mass.
3. Compute the takeoff mass by numerically solving the ordinary differential equation

$$\frac{dm}{dt} = f(m, t) \tag{1}$$

   which indicates that the rate at which the mass of the aircraft changes is equal to the rate of fuel consumption, which in turn is a function of the current mass of the aircraft and the current time in the mission. At each time step, the simulator's aerodynamic model is used to compute the current drag, and the simulator's propulsion model is used to compute the fuel consumption required to generate the thrust which will compensate for the current drag.

To test the techniques described in this paper, we used a twelve-dimensional design space in which the optimizer varied the following aircraft design parameters over a continuous range of values:

1. engine size
2. wing area
3. wing aspect ratio

4. fuselage taper length (how "pointed" the fuselage is)
5. effective structural thickness over chord (a nondimensional measure of wing thickness)
6. wing sweep over design mach angle (a nondimensional measure of wing sweep)
7. wing taper ratio (wing tip chord divided by wing root chord)
8. fuel annulus width (the amount of space left in the fuselage for fuel)
9. nozzle convergent flap length (l_c)
10. nozzle divergent flap length (l_d)
11. nozzle external flap length (l_e)
12. nozzle radius at station 7 (r_7)

This set of twelve design variables was chosen in collaboration with an aircraft industry design expert.* However, in these experiments we omitted discrete parameters, such as number of engines, which did not fit well with our continuous nonlinear programming search method. Since there are only a small number of choices, in practice our continuous design methodology could simply be repeated several times using different numbers of engines, and the best of these four or five designs could be chosen. A more general approach would be the use of mixed integer/continuous programming techniques as a search procedure, but that would require significant additional research.

Our optimizations focused on two aspects of the aircraft: the airframe, which is described by the first eight parameters (see Figure 1), and the exhaust nozzle, which is described by the last four parameters. Figure 2 shows the class of nozzles supported by the current system, the axisymmetric scheduled convergent-divergent exhaust nozzles often found in supersonic aircraft.[30] In Figure 2, r_{10}, r_e, and r_7 are fixed radii, and r_8 and r_9 are radii which are mechanically varied during aircraft operation. r_{10} is the outer radius of the engine to which the nozzle is attached, r_e is the radius of the duct leaving the engine, r_7 is the radius of the duct at the beginning of the movable convergent section of the nozzle, r_8 is the (variable) radius of the nozzle throat, and r_9 is the (variable) nozzle exit radius. Mechanically, this nozzle is a four-bar linkage, with three movable links labeled in Figure 2 by their lengths l_c, l_d, and l_e. During aircraft operation, the linkage is moved to change r_8 so that the cross-sectional area at the nozzle throat will produce desired engine performance. Since a four-bar linkage with three movable links has one degree of freedom, setting r_8 also sets r_9. In the experiments described in this paper, we allowed the optimizer to vary l_c, l_d, l_e, and r_7.

Our aircraft simulator supports two different ways of simulating the nozzle, which we use as two different levels of abstraction in our multilevel optimizations. The first method takes as input the parameters describing the flap lengths within the nozzle, and simulates the actual operation of the nozzle throughout the mission. The second method uses what is known as an "ideal nozzle." This method does not actually simulate the movement of the flaps within the nozzle, but instead assumes that the nozzle will always produce a certain efficiency. This abstraction of the model allows faster simulation, and does not require the nozzle flap lengths to be input to the simulator. A complete mission simulation requires about 1/2 second of CPU time on a DEC Alpha 250 4/266 desktop workstation when using the four-bar nozzle, and about 1/4 second when using the ideal nozzle.

The optimizations described in this paper were performed subject to a set of constraints. Table 4 lists the constraints, along with the values of the constraint functions at the point which we believe is the global optimum. The nozzle geometry bound constraints require, for example, that the specified nozzle flaps be con-

*Dr. Gene Bouchard of Lockheed

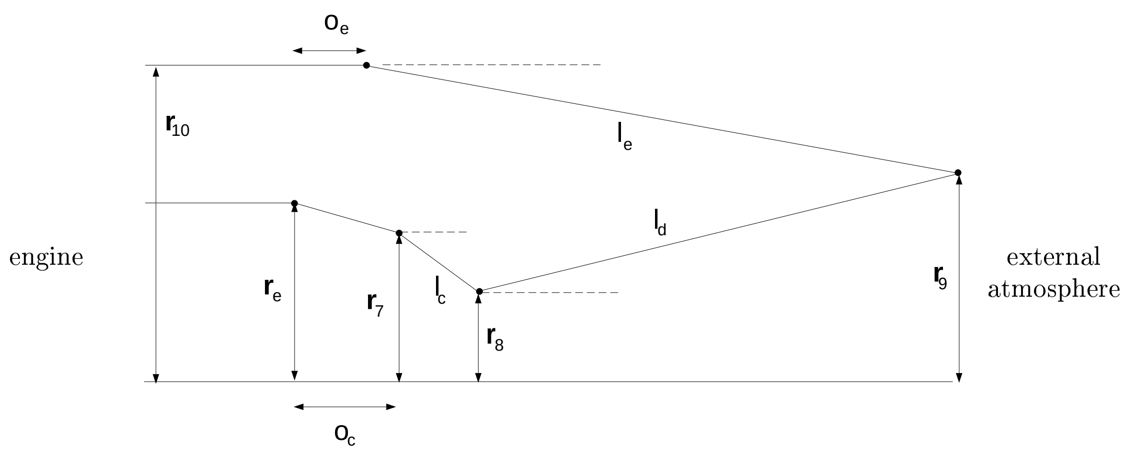

Figure 2: Axisymmetric convergent-divergent exhaust nozzle (flow from left to right)

nectable. The table bound constraints require that the simulator not have to extrapolate outside the tables of experimental data which it uses. The aerodynamic bounds require, for example, that the lift coefficient required to fly the specified design over the specified mission not exceed one. There is a sanity check to make sure that the work performed by the actuators in the nozzles is positive. Finally, the passenger constraint requires that there be enough room in the plane for the specified number of passengers. A more complete description of the constraints can be found in the appendix.

7 The Design Associate and Modeling/Simulation Associate

Figure 3 shows a block diagram of our automated conceptual design system. The design system has two major components: the Design Associate (DA), which searches the space of candidate designs, and the Model/Simulation Associate (MSA), which the DA uses to evaluate the quality of candidate designs. Unlike the discrete search spaces more commonly studied by AI researchers, the search space for the aircraft conceptual design problem involves design variables such as wing area or aspect ratio which can be varied continuously throughout an interval of possible values. To search this space, the DA uses a constrained nonlinear numerical optimizer, which varies the set of continuous design variables to minimize a nonlinear objective function subject to a set of nonlinear equality and inequality constraints. As mentioned previously, for the experiments reported in this article, the nonlinear objective function to be minimized is the takeoff mass required for a particular candidate aircraft design to fly a particular mission, and the optimizer used is CFSQP. The Model/Simulation Associate computes the values of the objective function and the design constraints, as well as a set of *model constraints* which are used to prevent the optimizer from going into regions of the search space that violate the simulator's underlying assumptions.[31] All of the constraints are further described in the appendix.

The data flow in Figure 3 is as follows:

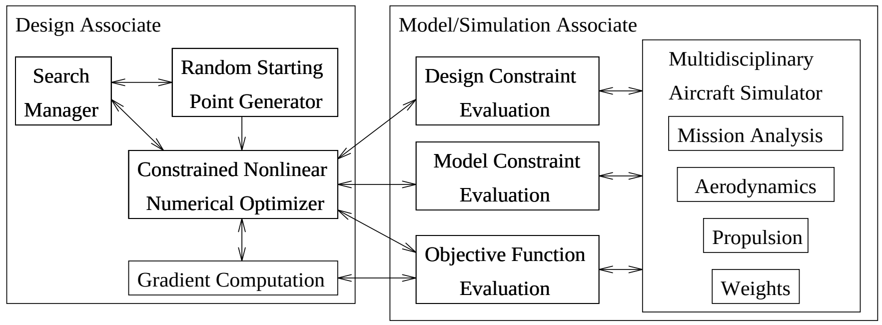

Figure 3: Automated design system block diagram

- The search manager (in conjunction with the random starting point generator) passes to the constrained nonlinear optimizer a design represented as a vector of real numbers, the values of the design variables. The optimizer uses this initial design as a starting point and later passes back an improved design using the same representation.

- The constrained nonlinear optimizer (directly or via the gradient computation module) passes to the evaluation modules a design, again represented as a vector of values of the design variables. The design constraint evaluation module passes back a vector of real numbers representing the values of the design constraints, the model constraint module does the same for model constraints, and the objective function evaluation module passes back a scalar value for design quality, which however is only meaningful if all the constraints are satisfied.

- The evaluation modules pass on to the simulator the design passed to them, and the simulator passes back a complete set of simulation results from which each evaluation module then extracts the data it needs.

In order to handle unevaluable points (i.e., points whose objective function cannot be evaluated by the simulator because, for example, the simulator crashes or returns an error message), the DA includes methods for "intelligent" gradient computation. The gradients used by CFSQP are computed by using a set of rules that specify how to compute gradients with reasonable accuracy in the presence of unevaluable points. For example, if the DA evaluates three candidate designs in order to compute a component of the gradient using a central difference formula, and if one of the points is unevaluable, then the DA ignores the unevaluable point and uses the other two points in a forward difference formula. The DA's rules for gradient computation are described in our previous work.[32] In addition, we have arranged for the line searches in CFSQP to terminate when they encounter unevaluable points.

8 Experimental Results

We made the following hypotheses:

1. Using an appropriately selected two-level decomposition for optimization will produce better optimization performance (lower CPU cost for the same probabil-

ity of getting a given design quality) than using one-level optimization.

2. Using an appropriately selected three-level decomposition for optimization will produce better optimization performance than using the two-level decomposition for optimization.

3. The same multilevel decompositions will produce good optimization performance for different missions.

To test our hypotheses, we performed optimizations using one, two, or three levels of decomposition, and then compared the results.

8.1 One-level optimization

First we tried doing optimization without the use of any multilevel techniques. We used the four-bar nozzle simulator, and used CFSQP to optimize in the search space defined by all twelve design parameters. Because this search space has many apparent local optima, we used a 100-point random multistart within the box of Table 2 to attempt to find the global optimum. The first curve in Figure 4 shows the estimated cost (number of simulations) of having a 99% chance of getting within various fractions of the takeoff mass that we believe to be the global optimum using this method. (We do not know with certainty what the global optimum is. Finding the global optimum of an arbitrary nonlinear function is undecidable.[28] However, we have performed many optimizations from different random starting points, and a large number of them have converged to the same point. We call this point "the apparent global optimum.") This estimate is computed by multiplying the average cost per optimization times $\log(1-P_{\mathrm{desired}})/\log(1-P_{\mathrm{success}})$, where P_{desired} is the desired probability of getting within the specified fraction of the apparent global optimum (99% in this case) and P_{success} is the probability of any single optimization getting within the specified fraction of the apparent global optimum (which we estimate using the fraction of our 100 optimizations that got within this fraction of the apparent global optimum). (The formula can be derived as follows: $(1-P_{\mathrm{success}})$ is the probability that a single optimization will **not** find the global optimum, so $(1-P_{\mathrm{success}})^n$ is the probability that **none** of n optimizations will find the global optimum, and thus $(1-(1-P_{\mathrm{success}})^n)$ is the probability that at least one of n optimizations **will** find the global optimum. To find the cost of P_{desired}, a given desired probability of finding the global optimum, solve

$$P_{\mathrm{desired}} = 1-(1-P_{\mathrm{success}})^n \qquad (2)$$

for n, which gives

$$n = \log(1-P_{\mathrm{desired}})/\log(1-P_{\mathrm{success}}) \qquad (3)$$

and finally multiply n by the average cost per optimization. Note: the computed value of n is not necessarily an integer, so a more precise calculation would round n up to the nearest integer.)

8.2 Two-level optimization

Since the one-level optimization was unacceptably expensive, we attempted to reduce the optimization cost by decomposing the search space into two levels. As our first level, we used the eight airframe parameters, and the ideal nozzle simulator. As our second level, we used the four nozzle parameters, and the four-bar nozzle simulator. CFSQP quickly found the point that we believe to be the optimum in the eight-dimensional airframe space, which was encouraging. The design parameters of the apparent optimum in the eight-dimensional space are shown in Table 3. We then fixed the values of the eight parameters

Design Parameter	low	high
engine size	0.5	3
wing area	139.4 m^2 (1500 ft^2)	1254.2 m^2 (13 500 ft^2)
wing aspect ratio	1	2
fuselage taper length	30.48 m (100 ft)	60.96 m (200 ft)
effective structural thickness over chord	1	5
wing sweep over design mach angle	1	1.45
wing taper ratio	0	0.1
fuel annulus width	0	1.219 m (4 ft)
nozzle convergent flap length	0.0762 m (3 in)	1.219 m (48 in)
nozzle divergent flap length	0.2286 m (9 in)	3.048 m (120 in)
nozzle external flap length	0.6096 m (24 in)	3.048 m (120 in)
nozzle radius at station 7	0.0254 m (1 in)	2.540 m (100 in)

Table 2: Subset of design space explored. See Section 6 for a description of each design parameter.

Design Parameters	
engine size	1.532
wing area	432.2 m^2 (4652 ft^2)
wing aspect ratio	1.570
fuselage taper length	36.97 m (121.3 ft)
effective structural thickness over chord	3.002
wing sweep over design mach angle	1.158
wing taper ratio	0
fuel annulus width	0

Table 3: Best design found in the eight-dimensional space, for the first mission.

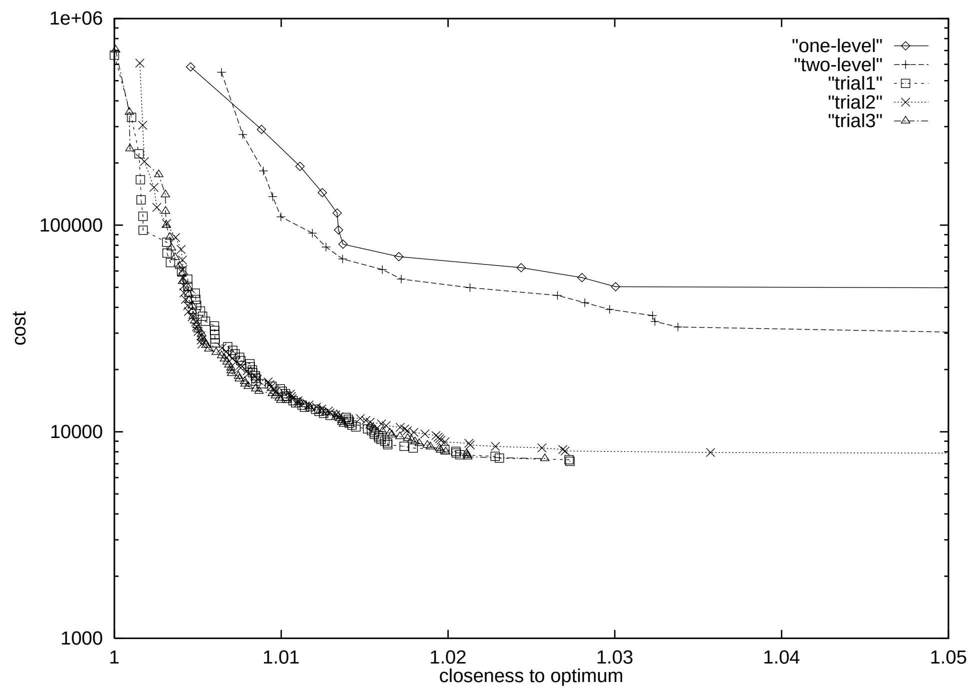

Figure 4: Performance of multilevel strategies for the first mission. Optimization performance increases as one moves down (lower cost) and to the left (closer to apparent optimum). The cost shown is the estimated number of simulations needed to have a 99% chance of getting within the specified fraction of the optimum. The three curves labeled “trial1,” “trial2,” and “trial3” represent three trials of the three-level method.

at their optimized values from the first level, and attempted to find the optimum in the nozzle space, using random multistart. After performing 1000 optimizations from random starting points, CFSQP failed to find even a single feasible point, so we declared this particular multilevel strategy to be a failure. We determined that the airframe designed in the first level, which had been designed using an ideal nozzle in the simulator, was only suitable for use with an ideal nozzle, so it was not possible to design a four-bar nozzle that would work with this airframe.

To circumvent this problem, we allowed CFSQP to vary all twelve design parameters in the second level. We performed a 5-point random multistart at the first (8-dimensional) level using the ideal nozzle, followed by a 100-point random multistart optimization at the second (12-dimensional) level using the four-bar nozzle, where each optimization starting point had the eight airframe parameters set at their optimized values from the first level, and had randomly generated values of the four nozzle parameters. The second curve in Figure 4 shows the estimated cost of having a 99% chance of getting within various fractions of the apparent optimum using this method. Each point in this curve is based on the cost of doing the 5-point 8-dimensional multistart, plus an n-point 12-dimensional multistart, for a value of n corresponding to the cost.

Two-level optimization resulted in roughly a twofold reduction in the cost of getting within a certain distance of the apparent optimum, supporting our first hypothesis. This improvement resulted from a combination of a reduced cost per optimization, and a smaller number of optimizations needed to find the apparent global optimum.

Design Parameters:	
engine size	1.462
wing area	418.8 m^2 (4508 ft^2)
wing aspect ratio	1.557
fuselage taper length	36.88 m (121.0 ft)
effective structural thickness over chord	2.978%
wing sweep over design mach angle	1.159
wing taper ratio	0
fuel annulus width	0
nozzle convergent flap length	0.3889 m (15.31 in)
nozzle divergent flap length	1.792 m (70.54 in)
nozzle external flap length	2.578 m (101.48 in)
nozzle radius 7 length	0.3716 m (14.63 in)
Objective Function:	
Takeoff Mass	162 200 kg
Nozzle geometry bounds:	
0.0-z7	-0.3971
r6-r10	-0.09784
r7-r10	-0.2806
z10-(z7+cl+dl)	-2.578
(r7-cl)-r6	-0.572
camin-camax	-1.272
el-elmax	-0.01806
elmin-el	-1.328
minRadius8-idealThroatRadius	-0.1240
idealThroatRadius-maxRadius8	-0.0001676
Table bounds:	
ECD lbte	-5.699
ECD ubte	-42.12
rae x min	-2.175
rae x max	-1.825
rae y min	-1.549
rae y max	-8.451
CA x min	-0.8136
CA x max	-98.03
CA y min	-4.121
CA y max	-35.71
CV x min	-0.8136
CV x max	-98.03
CV y min	-3.871
CV y max	-35.70
CB x min	-3.381
CB x max	-11.62
CB y max	-1.000
CB z min	-0.4876
CB z max	-0.5124
Aerodynamic bounds:	
wing-loading bound	-149.7 kg
fuel mass constraint	0
Lift coef-1	0.0
0.0-wing sweep	-1.214 rad
wing sweep-pi/2	-0.3569 rad
Sanity check:	
0.0-4barWork	-191015
Design Constraint:	
passenger constraint	-2

Table 4: Best design found for mission of Table 1. Negative values of constraint functions indicate that the constraints are satisfied. See the appendix for a description of the constraint functions.

8.3 Three-level optimization

Two-level optimization significantly reduced the cost of finding the apparent optimum in the 12-dimensional airframe/nozzle space, confirming our first hypothesis. However, we believed that further improvements in optimization performance would be possible if we could allow CFSQP to optimize the nozzle without at the same time optimizing all of the airframe parameters. We decided to try a new strategy for the second level: letting CFSQP optimize the nozzle parameters, and just one airframe parameter. We chose wing area as the one airframe parameter to optimize in the second level, because we believe that it is the most important airframe parameter. One can think of wing area alone as an abstraction of the entire airframe, to be used while optimizing the nozzle, much as the ideal nozzle is used as an abstraction of the four-bar nozzle while optimizing the airframe. Each run at the second level started with the eight airframe parameters set to their optimized values from the first level, and with the four nozzle parameters set randomly, and then did an optimization in the five-dimensional space defined by the four nozzle parameters and wing area. (For each starting point, we kept wing area at its optimized value, rather than setting it to a random value, because we believe that the optimized value, although not optimal, should be better than a random value.) We knew that optimizing in this space would not allow the optimizer to get exactly to the global optimum, so we added a third level in which the optimizer is allowed to vary all twelve design parameters. The third level was run each time that the level two optimization ended at a feasible point. The three curves in Figure 4 labeled "trial1," "trial2," and "trial3" show the estimated cost of having a 99% chance of getting within various fractions of the apparent optimum using the three-level method. Each of these curves is based on a different 5-point multistart in the 8-dimensional space of level one, followed by an n-point 5-dimensional multistart (for various values of n corresponding to the costs) at level two, followed by a third level in which there is a 12-dimensional optimization from each of the feasible apparent optima of level two. We did three trials of the three-level method to see if the results would vary significantly based on what happens in the random multistart of level one; the graph in Figure 4 shows that there is not much variation.

Using the three-level method provided roughly an order of magnitude reduction in cost compared with the two-level method, confirming our second hypothesis. We believe that there are three reasons for this speedup. The first is that computing the gradient is less expensive in the five-dimensional space. The second is that in the two-level method, when CFSQP is started from a point in which eight of the design parameters are nearly optimal and the other four are set to random values, it does not know that the eight airframe parameters are near their globally optimal values, so it initially changes the airframe parameters to make the airframe more appropriate for the suboptimal nozzle, and later has to change them back as the nozzle becomes closer to optimal, resulting in the need to perform more iterations. The third reason is that when doing 5-dimensional optimizations, CFSQP has a higher success rate at finding a feasible point than when doing 12-dimensional optimizations. In the two-level method, 30 of the 100 12-dimensional optimizations succeeded at finding a feasible point, while in the three-level method, an average of 76 out of the 100 five-dimensional optimizations succeeded in finding a feasible point. We believe that the reason for this higher success rate is that the constraint functions have fewer apparent local optima in the 5-dimensional space than they do in the 12-dimensional space.

Phase	Mach	Altitude m	ft	Duration (min)	comment
1	0.227	0	0	5	"takeoff"
2	0.85	12 192	40 000	50	subsonic cruise (over land)
3	2.0	18 288	60 000	225	supersonic cruise (over ocean)

capacity: 70 passengers.

Table 5: Another mission specification

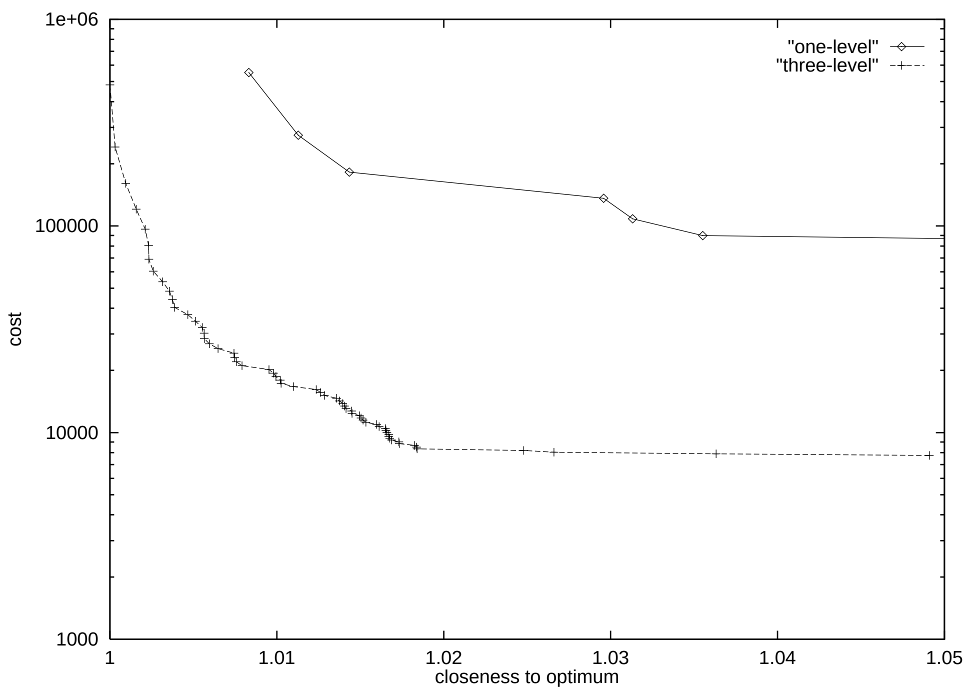

Figure 5: Performance of multilevel strategies for the second mission.

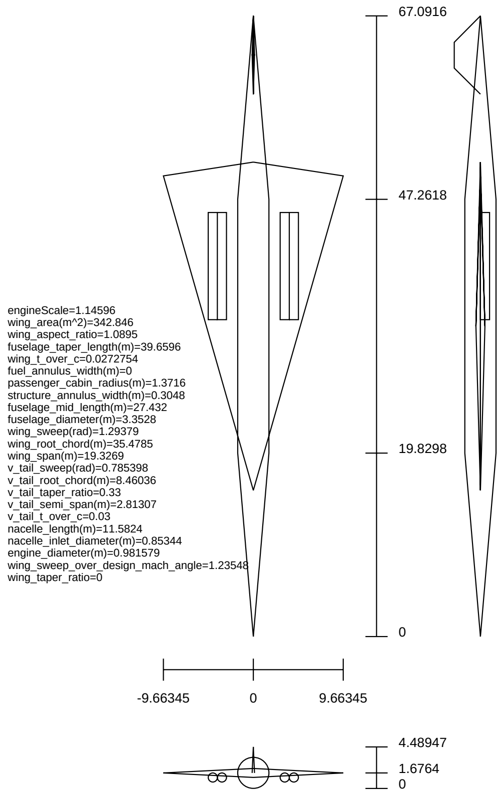

Figure 6: Supersonic transport aircraft designed by our system for the second mission (dimensions in meters)

Design Parameters:	
engine size	1.155
wing area	336.0 m^2 (3617 ft^2)
wing aspect ratio	1.091
fuselage taper length	39.41 m (129.3 ft)
effective structural thickness over chord	2.673%
wing sweep over design mach angle	1.232
wing taper ratio	0
fuel annulus width	0
nozzle convergent flap length	0.7176 m (28.25 in)
nozzle divergent flap length	1.279 m (50.34 in)
nozzle external flap length	2.398 m (94.40 in)
nozzle radius 7 length	0.3782 m (14.89 in)
Objective Function:	
Takeoff Mass	132 100 kg
Nozzle geometry bounds:	
0.0-z7	-0.4095
r6-r10	-0.1053
r7-r10	-0.2196
z10-(z7+cl+dl)	-2.406
(r7-cl)-r6	-0.8320
camin-camax	-0.5552
el-elmax	-0.02223
elmin-el	-1.172
minRadius8-idealThroatRadius	-0.2032
idealThroatRadius-maxRadius8	0
Table bounds:	
ECD lbte	-14.46
ECD ubte	-9.046
rae x min	-1.789
rae x max	-2.211
rae y min	-2.005
rae y max	-7.995
CA x min	-0.9595
CA x max	-97.75
CA y min	-6.426
CA y max	-33.10
CV x min	-0.9595
CV x max	-97.75
CV y min	-6.176
CV y max	-33.10
CB x min	-1.634
CB x max	-11.98
CB y max	-1.000
CB z min	-0.5166
CB z max	-0.4834
Aerodynamic bounds:	
wing-loading bound	-143.9 kg
fuel mass constraint	0
Lift coef-1	0
0.0-wing sweep	-1.290 rad
wing sweep-pi/2	-0.2803 rad
Sanity check:	
0.0-4barWork	-233402
Design Constraint:	
passenger constraint	-2

Table 6: Best design found for mission of Table 5. Negative values of constraint functions indicate that the constraints are satisfied. See the appendix for a description of the constraint functions.

8.4 Another mission

To test the effect of the mission on our results, we repeated the experiments for another mission — the mission of Table 5. We compared the single-level method with the three-level method. The results are shown in Figure 5. The best design found for this mission is shown in Table 6 and Figure 6. We again obtained an order of magnitude reduction in cost using the multilevel method, confirming our third hypothesis.

9 Analysis

We believe that the full twelve-dimensional space has a large number of apparent local optima, so that finding the apparent global optimum requires a large number of random multistarts. The two-level strategy reduces the cost by getting eight of the twelve parameters close to their optimal values in Level 1, so that fewer random multistarts are needed in the twelve dimensional space. This point is illustrated by noting that the optimized design parameters in the eight-dimensional space (see Table 3) are close to the optimized values of these parameters in the twelve-dimensional space (see Table 4), compared with the size of the box (see Table 2). The three-level strategy provides a further improvement by getting all twelve parameters near their optimal values in Levels 1 and 2, so that fewer twelve-dimensional optimizations are needed in Level 3.

We showed our designs to our aircraft industry design expert and he reported that the designs themselves and our methodology both seemed reasonable from his point of view. However, funding did not permit generating designs for these missions using today's industry techniques which could then have been compared with the designs our experiments generated. The expert did inform us that this sort of parametric design at a system level of abstraction is widely used in the aircraft industry and is considered quite productive.

10 Limitations and Future Work

One might ask whether the multilevel technique is applicable to design problems outside the aircraft domain. We have formulated (but not yet tested) multilevel techniques for two other domains: the design of racing yachts of the type used in the America's Cup race, and the design of a supersonic missile inlet. So far we have only performed single-level optimizations in each of these domains.[28]

In the racing yacht design domain,[33,34] we could use two levels of representation and analysis for the keel. At the first level, the keel would be analyzed using the following simple algebraic formula for effective draft (where D is maximum draft, and A_{ms} is the cross-sectional area of the hull at mid-ship):

$$T_{eff} = 0.92\sqrt{D^2 - \frac{2A_{ms}}{\pi}} \tag{4}$$

At this level, the keel would be represented using a small set of parameters that have an effect on the formula, or on other quantities computed by the yacht simulator, such as surface area or displacement. This small set of parameters would include the keel's height and taper ratio.

At the second level, the keel would be analyzed using PMARC, a panel method. Since PMARC is sensitive to the shape of the keel, the keel would be represented using a B-Spline surface. The cost of analyzing the keel with PMARC is orders of magnitude greater than the cost of evaluating the algebraic formula, so it would be potentially very beneficial to perform most of the optimization at the first level.

In the supersonic missile inlet design domain,[35,36] we have used an empirical code known as NIDA to analyze a missile inlet rapidly, and a computational fluid dynamics code known as GASP to analyze it with greater accuracy. Analyzing a single missile inlet with GASP takes about one CPU week, which makes it infeasible to perform optimizations with GASP using our current computational resources. We have instead performed optimizations with NIDA, and used GASP to verify the optimized designs. If we had greater computational resources available, we could perform inlet optimization at two levels, which would be likely to produce better designs than our current one-level NIDA optimization. The first level would be the same as our current optimizations — it would use NIDA for the analysis, and a nine-parameter design space that allows the optimizer to vary only those aspects of the inlet that are properly modeled by NIDA. The second level would use GASP for the analysis, and would have a higher-dimensional design space (possibly using splines) that allows the optimizer to make a wider range of changes to the shape of the inlet, since GASP is much more sensitive to the inlet's shape.

Note that these two applications of the multilevel paradigm are different from the airframe/nozzle domain application, in that the two levels do not decompose the search space. Instead, the first level is an abstraction of the second level. We hope that the simplified, lower-dimensional abstract level will have its global optimum close to the global optimum of the second level. In each case, the first level uses a simulator that is orders of magnitude faster than the more accurate simulator at the second level, so that we can perform numerous optimizations at the first level in less CPU time than it takes to do a single simulation at the second level. We hope that after finding the global optimum at the first level, we will be able to reach the global optimum at the second level using only one optimization, or perhaps using a small number of random multistarts. It may also be the case that the simplified search space of level one has fewer local optima than the full search space of level two. Whether the first level has fewer local optima or not would not be very important, however, since it would be possible to perform a large number of random multistarts at the first level at very little cost compared with the cost of performing an optimization at the second level.

It may be difficult to identify the appropriate simulator and search-space abstractions in still other domains. Automatically identifying such abstractions is an area for future research. Finally, the performance of our approach of performing optimization in the presence of many apparent local optima by using a gradient-based optimizer at multiple levels of abstraction needs to be compared with that of global methods such as genetic algorithms and simulated annealing. We may even find that it is possible to use these global methods at multiple levels of abstraction, for even better optimization performance.

11 Conclusion

Multilevel representations have been studied extensively by artificial intelligence researchers. We have presented a general method that utilizes the multilevel paradigm to attack the problem of performing multidiscipline engineering design optimization in the presence of many local optima. The method uses a multidisciplinary simulator at multiple levels of abstraction, paired with a multilevel search space. We demonstrated the effectiveness of this general method by testing it in the domain of conceptual design of supersonic transport aircraft, focusing on the airframe and the exhaust nozzle, and using sequential quadratic programming as the optimizer at each level. We found that using multilevel simulation and optimization can decrease the cost of design space search by an order of magnitude.

Acknowledgments

We thank our aircraft design expert, Gene Bouchard of Lockheed, for his invaluable assistance in this research. We also thank all members of the HPCD project, especially Don Smith, Keith Miyake, Khaled Rasheed, Brian Davison, and Tom Ellman. We thank Rob Ivester and Kathleen Romanik of NIST for reviewing this manuscript. This research was partially supported by NASA under grant NAG2-817 and is also part of the Rutgers-based HPCD (Hypercomputing and Design) project supported by the Advanced Research Projects Agency of the Department of Defense through contract ARPA-DABT 63-93-C-0064. Mark Schwabacher was supported by a National Research Council Postdoctoral Research Associateship for a portion of the time that he worked on this article.

Disclaimer

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipment, instruments, or materials are identified in this report in order to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Appendix

A Constraints

Nozzle geometry bounds: The geometry of the nozzle's fourbar linkage must satisfy the following constraints:

NG1 $= 0 - z_7$. Nozzle geometry bound.

NG2 $= r_6 - r_{10}$. Nozzle geometry bound.

NG3 $= r_7 - r_{10}$. Nozzle geometry bound.

NG4 $= z_{10} - (z_7 + l_c + l_d)$. Nozzle geometry bound.

NG5 $= (r_7 - l_c) - r_6$. Nozzle geometry bound.

CA = <minimum angle to which convergent flap can move, while still maintaining a convergent-divergent configuration> − <maximum angle to which convergent flap can move, while still maintaining a convergent-divergent configuration>.

ELMAX = <length l_e of external nozzle flap> − <maximum length external nozzle flap could have, with the given values for the rest of the nozzle geometry, while still allowing the nozzle to be connected as a convergent-divergent nozzle>.

ELMIN = <minimum length external nozzle flap could have, with the given values for the rest of the nozzle geometry, while still allowing the nozzle to be connected as a convergent-divergent nozzle> − <length l_e of external nozzle flap>.

R8LOW = <smallest value r_8 can achieve with current geometry> − <smallest value for r_8 required during mission simulation>.

R8HIGH = <largest value for r_8 required during mission simulation> − <largest value r_8 can achieve with current geometry while maintaining a convergent-divergent configuration>.

Table bounds: The simulator must not extrapolate outside certain tables of experimental data:

ECD ubte = <maximum throttle required during mission simulation> − <maximum throttle setting allowed for engine>. If an impossibly high throttle is required to fly the mission, the simulation will continue using extrapolation, but the value of ETUB will indicate the extent to which the engine model assumptions are violated.

ECD lbte = <minimum throttle setting allowed for engine> − <minimum throttle required during mission simulation>.

rae: Similar to above — violation of bounds for a two-dimensional table of experimental data on supersonic drag.

CA: violation of bounds for a two-dimensional table of experimental data on nozzle angularity thrust loss.

CV: violation of bounds for a two-dimensional table of experimental data on nozzle friction velocity/thrust loss.

CB: violation of bounds for a two-dimensional table of experimental data on nozzle boattail (external) drag.

Aerodynamic bounds: These constraints ensure that the aircraft is capable of flying the specified mission:

WLUB = <maximum wing loading during mission simulation> − <maximum wing loading simulator can validly model>.

FM = <fuel mass that current candidate design requires to complete mission> − <fuel mass that can be stored in available volume for current candidate design>.

STALL = <maximum lift coefficient during mission simulation> − <maximum lift coefficient simulator can validly model>. The simulator assumes wings won't stall, and this constraint function computes how well that assumption is satisfied.

wing sweep: Wing sweep angle must be between 0 and $\pi/2$.

Sanity check: This constraint prevents a certain type of insane result:

4barWork: The work performed by the actuators in the fourbar linkage must be positive.

Design constraint: This constraint ensures that the aircraft satisfies a particular constraint specified by the engineer:

PASS = <passenger capacity required for the mission> − <passenger capacity available with current design variables>.

References

[1] Sacerdoti, E. D. "Planning in a Hierarchy of Abstraction Spaces". *Artificial Intelligence*, Vol. 5, pp. 115 – 135, 1974.

[2] Simon, H. *The Sciences of the Artificial.* MIT Press, Cambridge, MA, 1981.

[3] Sobieszczanski-Sobieski, J. "A Linear Decomposition Method for Large Optimization Problems—Blueprint for Development". Technical Report NASA TM-83248, National Aeronautics and Space Administration, 1982.

[4] Sobieszczanski-Sobieski, J., James, B. B., and Dovi, A. R. "Structural Optimization by Multilevel Decomposition". *AIAA Journal*, Vol. 23, No. 11, pp. 1775–1782, November 1985.

[5] Rogers, J. L. "A Knowledge-Based Tool for Multilevel Decomposition of a Complex Design Problem". NASA Technical Paper 2903, Langley Research Center, National Aeronautics and Space Administration, Hampton, VA, 1989.

[6] Ellman, T. and Schwabacher, M. "Abstraction and Decomposition in Hill-climbing Design Optimization". Technical Report CAP-TR-14, Department of Computer Science, Rutgers University, New Brunswick, NJ, January 1993. ftp://ftp.cs.rutgers.edu/pub/technical-reports/cap-tr-14.ps.Z.

[7] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. *Numerical Recipes*. Cambridge University Press, New York, NY, 1986.

[8] Powell, D. and Skolnick, M. "Using genetic algorithms in engineering design optimization with non-linear constraints". In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 424–431, Univeristy of Illinois at Urbana-Champaign, July 1993. Morgan Kaufmann, Los Altos, CA.

[9] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass., 1989.

[10] Gill, P. E., Murray, W., and Wright, M. H. *Practical Optimization*. Academic Press, London ; New York, 1981.

[11] Peressini, A. L., Sullivan, F. E., and Uhl, Jr., J. J. *The Mathematics of Nonlinear Programming*. Springer-Verlag, New York, 1988.

[12] Vanderplaats, G. N. *Numerical Optimization Techniques for Engineering Design : With Applications*. McGraw-Hill, New York, 1984.

[13] Moré, J. J. and Wright, S. J. *Optimization Software Guide*. SIAM, Philadelphia, PA, 1993.

[14] Ellman, T., Keane, J., and Schwabacher, M. "Intelligent Model Selection for Hill-climbing Search in Computer-Aided Design". In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 594–599, Washington, DC, 1993. MIT Press, Cambridge, MA.

[15] Bouchard, E. E., Kidwell, G. H., and Rogan, J. E. "The Application of Artificial Intelligence Technology to Aeronautical System Design". In *AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting*, Atlanta, Georgia, September 1988. AIAA Paper 88-4426.

[16] Hoeltzel, D. and Chieng, W. "Statistical Machine Learning for the Cognitive Selection of Nonlinear Programming Algorithms in Engineering Design Optimization". In *Advances in Design Automation*, Boston, MA, 1987.

[17] Orelup, M. F., Dixon, J. R., Cohen, P. R., and Simmons, M. K. "Dominic II: Meta-Level Control in Iterative Redesign". In *Proceedings of the National Conference on Artificial Intelligence*, pages 25–30, St. Paul, MN, 1988. MIT Press, Cambridge, MA.

[18] Powell, D. *Inter-GEN: A Hybrid Approach to Engineering Design Optimization*. PhD thesis, Rensselaer Polytechnic Institute Department of Computer Science, Troy, NY, December 1990.

[19] Tong, S. S., Powell, D., and Goel, S. "Integration of Artificial Intelligence and Numerical Optimization Techniques for the Design of Complex Aerospace Systems". In *1992 Aerospace Design Conference*, Irvine, CA, February 1992. AIAA Paper 92-1189.

[20] Tong, S. S. "Coupling Symbolic Manipulation and Numerical Simulation for Complex Engineering Designs". In *International Association of Mathematics and Computers in Simulation Conference on Expert Systems for Numerical Computing*, Purdue University, West Lafayette, IN, 1988.

[21] Gage, P. *New Approaches to Optimization in Aerospace Conceptual Design*. PhD thesis, Stanford University, Stanford, CA, 1994.

[22] Gage, P., Kroo, I., and Sobieski, I. "Variable-Complexity Genetic Algorithm for Topological design". *AIAA Journal*, Vol. 33, No. 11, pp. 2212–2217, 1995.

[23] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I. "Multidisciplinary Optimization Methods for Aircraft Preliminary Design". In *Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City, Florida, September 1994. AIAA Paper 94-4325.

[24] Bramlette, M., Bouchard, E., Buckman, E., and Takacs, L. "Current applications of genetic algorithms to aeronautical systems". In *Proceedings of the Sixth Annual Aerospace Applications of Artifical Intelligence Conference*, October 1990.

[25] Sobieszczanski-Sobieski, J. and Haftka, R. T. "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments". In *34th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1996. AIAA Paper 96-0711.

[26] Lawrence, C., Zhou, J., and Tits, A. "User's Guide for CFSQP Version 2.3: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints". Technical Report TR-94-16r1, Institute for Systems Research, University of Maryland, College Park, MD, August 1995.

[27] Bazaraa, M., Sherali, H., and Shetty, C. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, New York, 1993.

[28] Schwabacher, M. "The Use of Artificial Intelligence to Improve the Numerical Optimization of Complex Engineering Designs". Technical Report HPCD-TR-45, Department of Computer Science, Rutgers University, New Brunswick, NJ, October 1996. Ph.D. Thesis. http://www.cs.rutgers.edu/~schwabac/thesis.html.

[29] Gelsey, A. and Smith, D. "Computational Environment for Exhaust Nozzle Design". *Journal of Aircraft*, Vol. 33, No. 3, pp. 470–476, 1996.

[30] Mattingly, J. D., Heiser, W. H., and Daley, D. H. *Aircraft Engine Design*. AIAA education series. American Institute of Aeronautics and Astronautics, New York, N.Y., 1987.

[31] Gelsey, A., Schwabacher, M., and Smith, D. "Using Modeling Knowledge to Guide Design Space Search". In Gero, J. and Sudweeks, F., editors, *Artificial Intelligence in Design '96*, pages 367–385. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[32] Schwabacher, M. and Gelsey, A. "Intelligent Gradient-Based Search of Incompletely Defined Design Spaces". *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 11, No. 3, pp. 199–210, June 1997.

[33] Schwabacher, M., Ellman, T., Hirsh, H., and Richter, G. "Learning to Choose a Reformulation for Numerical Optimization of Engineering Designs". In Gero, J. and Sudweeks, F., editors, *Artificial Intelligence in Design '96*, pages 447–462. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[34] Schwabacher, M., Ellman, T., and Hirsh, H. "Learning to Set Up Numerical Optimizations of Engineering Designs". *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Vol. 12, No. 2, , 1997. To appear.

[35] Zha, G.-C., Smith, D., Schwabacher, M., Rasheed, K., Gelsey, A., Knight, D., and Haas, M. "High Performance Supersonic Missile Inlet Design Using Automated Optimization". In *6th AIAA/NASA/USAF Multidisciplinary Analysis & Optimization Symposium*, Bellevue, WA, September 1996. AIAA Paper 96-4142.

[36] Zha, G.-C., Smith, D., Schwabacher, M., Rasheed, K., Gelsey, A., Knight, D., and Haas, M. "High Performance Supersonic Missile Inlet Design Using Automated Optimization". *AIAA Journal of Aircraft*, Vol. 34, No. 6, pp. 697–705, November 1997.